1 Mo Tu ~~We~~ Th Fr Sa Su

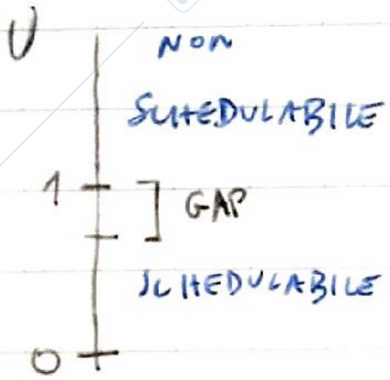
No. SISTEMI INF.

Date 09 . 10 . 19

	$T_i = D_i$	$D_i \leq T_i$
PRIORITÀ STATICA	RM (Rate Monotonic)	DM (Deadline Monotonic)
PRIORITÀ DINAMICA	EDF	EDF

Cambia nel tempo

Trattiamo i processi periodici



tutto ciò che posso fare negli statici posso farlo nei dinamici (cambiando ogni volta la priorità, suddividendolo quindi in tante priorità statica)

STATICA	DINAMICA
Prestazioni peggiori	Prestazioni maggiori
Meno calcolo, spazio computazionale	Occupa più spazio computazionale

RATE MONOTONIC

PROBLEMA: n PROCESSI PERIODICI, CON $D_i = T_i$

ALGORITMO: LA PRIORITÀ DI OGNI PROCESSO È $\frac{1}{T_i}$

PROPRIETÀ: È OTTIMO PER AMMISSIBILITÀ TRA TUTTI GLI ALGORITMI STATICI QUANDO

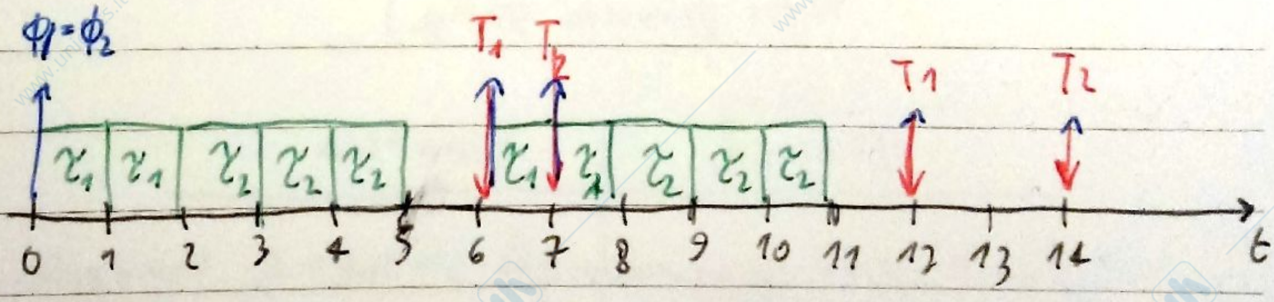
$$\phi_i = \phi_j \quad \forall i,j$$

↳ sono in fase

ES.

	τ_1	τ_2
ϕ_i	0	0
C_i	2	3
T_i	6	7

Stiamo assumendo che $D_i = T_i$



Possiamo considerarla, nel caso del rate monotonic una schedulazione gerarchica.

↳ ossia ha priorità massima quello che ha deadline relativa minore quindi schedutiamo prima questo processo, poi quelli con priorità più bassa

se $\phi_i = \phi_j \ \forall i,j$ ←

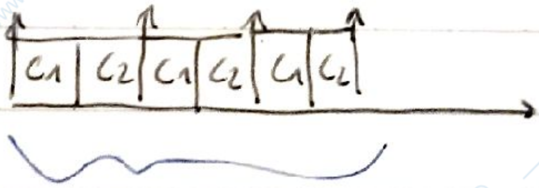
Provare a trovare problemi con τ_1, τ_2 tali che abbiano $U < 1$ e che non siano schedulabili con RM

es.

Se abbiamo

$T_1 = C_1$ $\quad T_1 = T_2$

$C_1 = \frac{T_1}{2}$



$C_2 = \frac{T_2}{2}$

ha sempre lo stesso PATTERN

→ ottimalità di RM

Dato un problema:

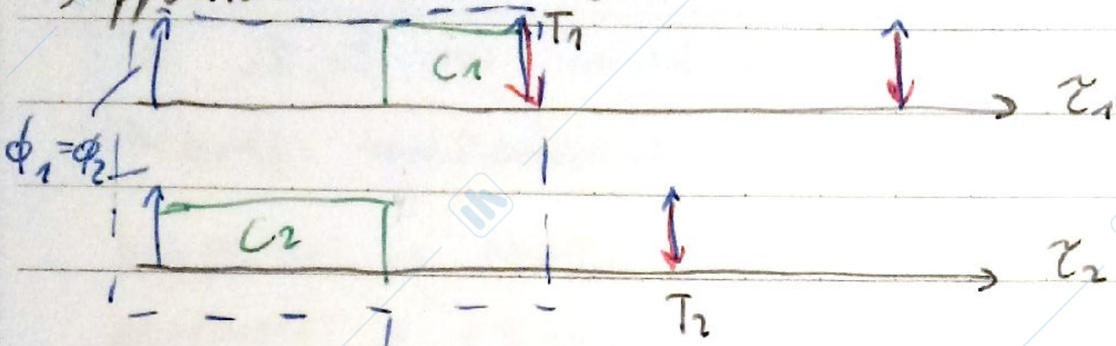
→ Se è schedulabile con un algoritmo statico diverso da RM, allora lo è anche con RM

quindi non esiste la possibilità che un algoritmo schedeli il mio problema, ma per questo RM fallisca

No. SISTEMI INF.

Date 09 · 10 · 19

Supponiamo di avere



$T_1 < T_2$

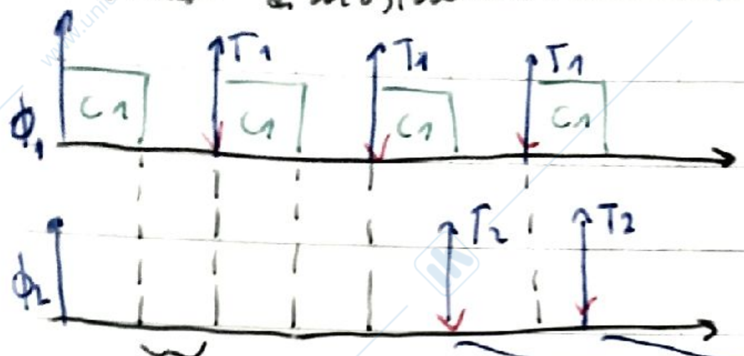
↳ SEZIONE CRITICA (1° PERIODO DI OGNI PROCESSO → scegliamo l'T_1 perché è più stringente)

↓

SE È SCHEDULABILE IN UNA SEZIONE CRITICA PER UN ALGORITMO ↳ È SEMPRE

↳ $\boxed{C_1 + C_2 \leq T_1}$

continuo dimostrazione:



Schedulo con RM

2 casi:

questa quantità di tempo deve essere almeno quando quanto C_2

1° → Facile da solo

2°

C_1 ancora in esecuzione

↓

$C_2 \leq$?

Dobbiamo considerare quante volte intero τ_1 viene eseguito

Lo chiamo $F = \left\lfloor \frac{T_2}{T_1} \right\rfloor$

questo simbolo significa che arrotondo verso il basso questo rapporto

Perciò lo spazio lasciato libero per C2 sarà:

CONDIZIONE DI SCHEDULABILITÀ → $C_2 \leq F \cdot T_1 - F \cdot C_1 = F(T_1 - C_1)$

2 | Mo Tu We ~~Th~~ Fr Sa Su

No. SISTEMI INF.

Date 10 · 10 · 19

Dobbiamo andare a trovare a una CONDIZIONE DI CASO per cui togliendo a T_2 il numero di volte intere che τ_1 viene eseguito otteniamo un pezzettino che è $\leq c_1$

$$\downarrow$$

$$T_2 - F \cdot T_1 \leq c_1 \leftarrow$$ CONDIZIONE DI CASO

Se vale $c_1 + c_2 \leq T_1$ e $T_2 - F \cdot T_1 \leq c_1$ $\Rightarrow c_2 \leq F(T_1 - c_1)$

$$c_2 \leq FT_1 - Fc_1$$

$$\hookrightarrow \boxed{c_2 + Fc_1 \leq FT_1}$$

$$\Uparrow$$

$F \geq 1$ per definizione $(T_2 > T_1)$

$$\boxed{c_1 + c_2 \leq T_1}$$

$$\hookrightarrow Fc_1 + Fc_2 \leq FT_1$$

quindi se questa è vera

allora questa è vera $(Fc_1 + Fc_2 \geq Fc_1 + c_2)$ ✓

Schedulabilità

$U \leq \bar{U} \longrightarrow$ sicuramente schedulabile con RM

$\Downarrow$

Vogliamo trovare questa soglia

Se cale $T_2 - FT_1 \leq C_1 \Rightarrow C_2 \leq FT_1 - FC_1$

Ci mettiamo nella condizione estrema

$\hookrightarrow C_2 = FT_1 - FC_1$

$\downarrow$

$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} = \frac{C_1}{T_1} + \frac{FT_1 - FC_1}{T_2} \rightarrow$$ U non dipende da C_2

$\vdots$

$$U = F\frac{T_1}{T_2} + \frac{C_1}{T_2}\left(\frac{T_2}{T_1} - F\right) \rightarrow$$ Si trova in un range $\bar{U} \leq U < 1$

$\downarrow$

voglio minimizzare U rispetto a tutti i suoi parametri

- $\rightarrow$ rispetto a C_1 $\quad \min\limits_{C_1} U = \min\limits_{C_1} \left(F\frac{T_1}{T_2} + \frac{C_1}{T_2}\overbrace{\left(\frac{T_2}{T_1} - F\right)}^{>0}\right)$

prendo $C_1 = T_2 - FT_1$ da $T_2 - FT_1 \leq C_1$

$$\hookrightarrow U = F\frac{T_1}{T_2} + \frac{(T_2 - FT_1)}{T_2}\left(\frac{T_2}{T_1} - F\right)$$

$$\ldots \quad U = \frac{T_1}{T_2}\left(F + \left(\frac{T_2}{T_1} - F\right)^2\right)$$

min rispetto a C_1

Chiamiamo

$$G = \frac{T_2}{T_1} - F$$

$$\hookrightarrow U = 1 - \frac{G(1-G)}{F+G}$$

- $\rightarrow$ rispetto a F $\quad \min\limits_F U = \min\limits_F \left(1 - \frac{\overbrace{G(1-G)}^{>0}}{F+G}\right)$

F deve essere il più piccolo possibile

$F \geq 1$

$G \in [0,1)$

$F = 1$

$$\hookrightarrow U = 1 - \frac{G(1-G)}{1+G}$$

- $\rightarrow$ rispetto a G $\quad \min\limits_G U = \min\limits_G \left(1 - \frac{G(1-G)}{1+G}\right)$

$$\frac{dU}{dG} = 0 \quad \frac{(2G-1)(1+G) - G^2 + G}{(1+G)^2} = \frac{G^2 + 2G - 1}{(1+G)^2} = 0$$

5

No. SISTEMI INF.

Date 10 10 19

$\rightarrow \quad G = \sqrt{2} - 1$

$\Rightarrow \quad \bar{U} = 2(\sqrt{2} - 1) \rightarrow$ SOGLIA DI SCHEDULABILITÀ DI RM CON $\underbrace{\text{2 PROCESSI}}$

con un numero di processi n

diventa

$\hookrightarrow \quad \bar{U} = n(\sqrt[n]{2} - 1) \rightarrow$ n PROCESSI (Bound di Liu & Layland)

Se osserviamo il limite di $\bar{U}$ per $n \to +\infty$:

$$\lim_{n \to +\infty} \left(n(\sqrt[n]{2} - 1) \right) \simeq 0{,}69$$

$\rightarrow$ Se $U \leq 0{,}69$ il nostro problema è $\wedge$ schedulabile

sicuramente

es. La condizione vale anche se i processi non sono in fase

①

	τ_1	τ_2
ϕ_i	1	0
T_i	5	7
C_i	2	4

$\bar{U} = 0{,}828$

$U = \frac{2}{5} + \frac{4}{7} = 0{,}97$

↓ non sappiamo se è schedulabile

②

	τ_1	τ_2	τ_3
ϕ_i	0	1	2
T_i	4	5	6
C_i	1	1	3

$\bar{U} = 0{,}779$

$U = \frac{1}{4} + \frac{1}{5} + \frac{3}{6} = 0{,}95$

↓ non sappiamo se schedulabile

③

	τ_1	τ_2
ϕ_i	1	0
T_i	5	7
C_i	1	2

$\bar{U} = 0{,}828$

$U = \frac{1}{5} + \frac{2}{7} = 0{,}485$

↓ schedulabile

se vogliamo arrotondarlo facciamolo verso il basso

Esiste anche un Bound di Bini & Buttazzo che dice

$$\prod_i (1 + U_i) \leq 2 \rightarrow \text{schedulabile}$$

Riconsiderando l'esempio precedente:

① $\left(1 + \frac{2}{5}\right)\left(1 + \frac{4}{7}\right) = 2{,}2 > 2$ → non si sa se sched.

② . . . . = $2{,}25 > 2$

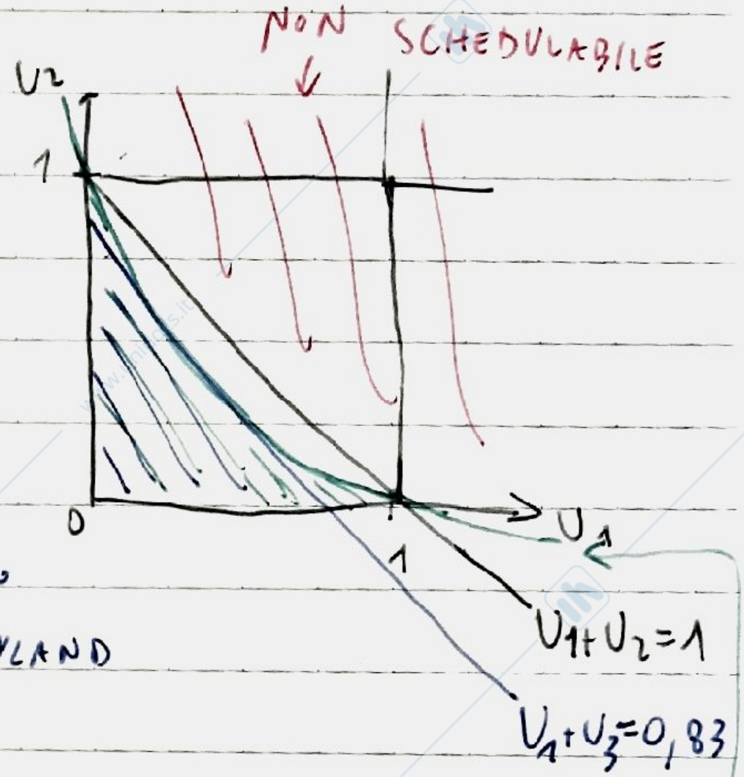
③ . . . . = $1{,}54 < 2$ → schedulabile

Esaminiamo la relazione fra i 2 bound visti precedentemente:

nel caso n=2

$U_1 = \frac{C_1}{T_1}$

$U_2 = \frac{C_2}{T_2}$



SCHEDULABILE SECONDO BOUND DI LYU & LAYLAND

SCHEDULABILE SECONDO BOUND DI BINI & BUTTAZZO

$$(1+U_1)(1+U_2) \leq 2$$

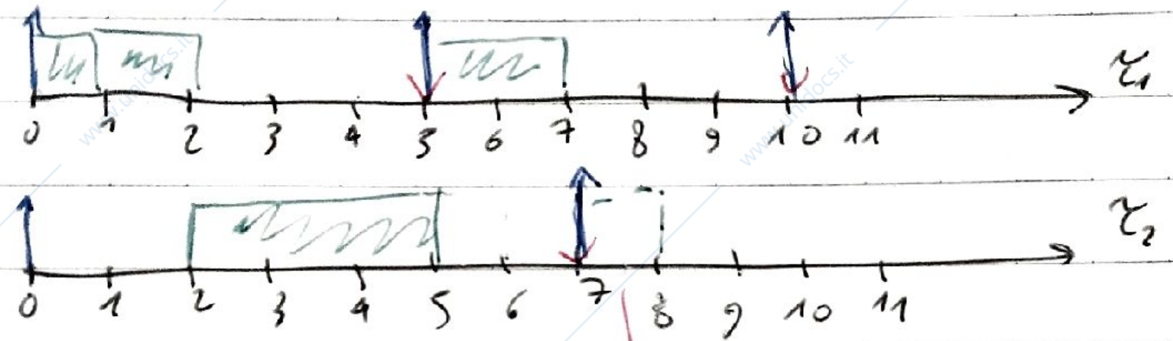
$$\hookrightarrow U_2 \leq \frac{2}{1+U_1} - 1$$

=> se schedulabile secondo Bound Lyu Layland => schedulabile secondo Bini & Buttazzo es. ($U<1$, ma non schedulabile con RM)

$U_1 = \frac{2}{5}$ $\quad U_2 = \frac{4}{7}$

	τ_1	τ_2
ϕ_i	0	0
C_i	2	4
T_i	5	7

$U = \frac{34}{35} < 1$



la deadline è violata

$\bar{U} = 0{,}83$

$\prod (1 + U_i) = \frac{11}{5} > 2$

Anche se $U<1$

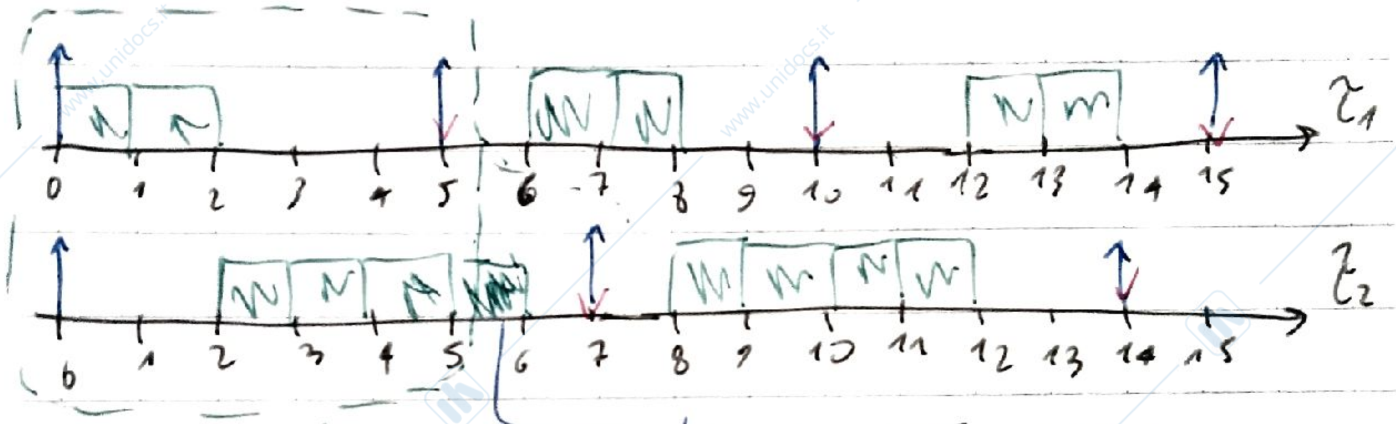
dobbiamo analizzare i bound

Se consideriamo le differenze tra gli algoritmi: statico (RM) e dinamico (EDF), osserviamo che EDF richiede un riordinamento dei processi.

alto costo computazionale

EDF (processi periodici)

Ogni attivazione di un processo periodico, viene gestita come se fosse un processo aperiodico indipendente.



ha priorità perché ha deadline più stringente

lo considero come un processo aperiodico singolo

Osserviamo che:

$$1 \geq \bar{U}_{EDF} \geq \bar{U}_{RM}$$

$$\| \quad 1$$

non ha zone "grigie" dove non sappiamo se schedula o no

$\Rightarrow$ Quindi se $U \leq 1 \Rightarrow$ EDF lo schedula

però alto costo computazionale

se $U > 1 \Rightarrow$ EDF non schedula

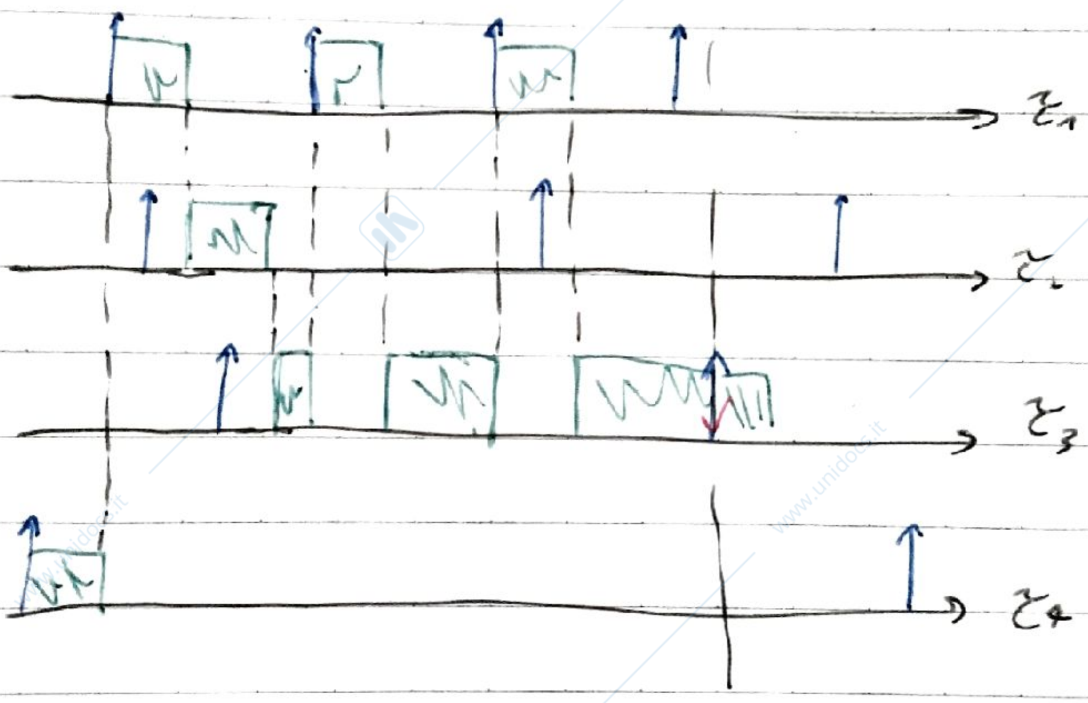
$\rightarrow$ non c'è algoritmo per sapere se schedula di EDF

Dimostrazione ($U \leq 1 \rightarrow$ SCHED. CON EDF)

Per assurdo non è possibile e assumiamo che $U \leq 1$ non schedulabile

Se non è schedulabile significa che

$\exists\, t_2$ in cui una deadline è violata



t_2 di violazione deadline

Vogliamo trovare l'intervallo $[t_1, t_2]$

t_1 a sinistra di t_2 — ossia quando il processore non sta calcolando niente

- non ci sono istanti di idle
- t_1 sia il più a sinistra possibile
- solo i processi con $d_i \leq t_2$ sono eseguiti

Considerando $[t_1, t_2]$ abbiamo che il tempo di computazione richiesto dai processi è

$$\sum_i \left\lfloor \frac{(t_2-t_1)}{T_i} \right\rfloor c_i$$

arrotondo verso il basso

Visto che per ipotesi non è schedulabile

$$\sum_i \left\lfloor \frac{(t_2-t_1)}{T_i} \right\rfloor c_i > t_2-t_1$$

Possiamo inoltre scrivere che:

$$\sum_i \left\lfloor \frac{(t_2-t_1)}{T_i} \right\rfloor c_i \leq \sum_i \frac{(t_2-t_1)}{T_i} c_i = (t_2-t_1) \underbrace{\sum_i \frac{c_i}{T_i}}_{U}$$

$\boxed{U \leq 1 \Rightarrow \text{SCHED.}}$

$$= (t_2-t_1) U$$

$$\rightarrow t_2-t_1 < \sum_i \left\lfloor \frac{(t_2-t_1)}{T_i} \right\rfloor c_i \leq (t_2-t_1) U$$

$$\hookrightarrow \cancel{t_2-t_1} < \cancel{(t_2-t_1)} U \quad \boxed{U > 1}$$

ASSURDO

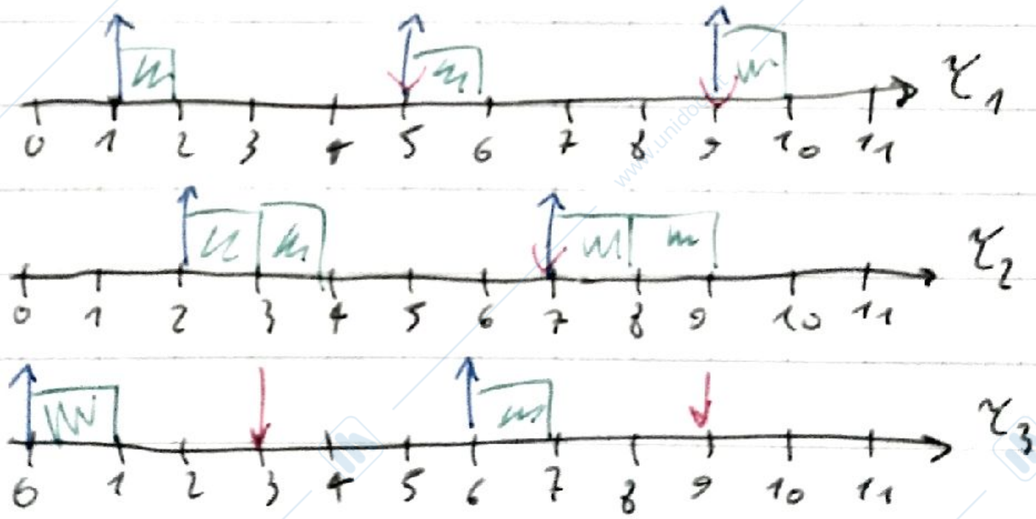
DM (Deadline monotonic)

Stiamo assumendo che $D_i \le T_i$

	τ_1	τ_2	τ_3
ϕ_i	1	2	0
T_i	4	5	6
c_i	1	2	1
D_i	4	5	3

È come RM, ma si ordina rispetto a D_i
(D_i più stringente)

es.



Oss.

Il Bound di Bini & Buttazzo non funziona

→ RTA (Response Time Analysis)
↳ Algoritmo che va a calcolare il tempo di risposta (R_i) per ogni processo sulla sua sezione critica.

tempo dall'attivazione alla terminazione

Sezioni in cui il ritardo è massimo → quindi se vale in questa zona vale per tutto il resto

$$R_i = c_i + \underbrace{I_i}$$

interferenza dovuta ai processi con priorità maggiore

Assumiamo che i processi siano ordinati $\tau_1 \dots \tau_n$ rispetto a D_i

arrotondamento verso l'alto

$$I_i = \sum_{j=1}^{i-1} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

Sto sommando su tutti i processi che hanno una deadline relativa $\leq D_i$

<u>Oss</u> Si arrotonda verso l'alto poiché il processo J ha priorità maggiore rispetto a i e viene sempre eseguito completamente prima che i sia eseguito

Quindi abbiamo:

$$R_i = C_i + \sum_{J=1}^{i-1} \left\lceil \frac{R_i}{T_J} \right\rceil C_J$$

Per risolvere questa equazione facciamo in modo iterativo:

$$R_i^{(0)} = C_i$$

$$I_i^{(0)} = \sum_{J=1}^{i-1} \left\lceil \frac{R_i^{(0)}}{T_J} \right\rceil C_J$$

$$\hookrightarrow R_i^{(k)} = C_i + I_i^{(k-1)}$$

Se $R_i^{(k)} = R_i^{(k-1)} \rightarrow$ FINE

↳ ALTRIMENTI
Calcola $I_i^{(k)}$

c).

	τ_1	τ_2	τ_3	τ_4
ϕ_i	0	0	0	0
T_i	4	5	6	11
C_i	1	1	2	1
D_i	3	4	5	10

$\tau_1 \to R_1 \Rightarrow R_1^{(0)} = C_1 = 1$

$I_1^{(0)} = 0$

$R_1^{(1)} = 1$ FINE

$\Rightarrow R_1 = 1$

$\tau_2 \to R_2 \Rightarrow R_2^{(0)} = C_1 = 1$

$I_2^{(0)} = \left\lceil \frac{R_2^{(0)}}{T_1} \right\rceil C_1 = \left\lceil \frac{1}{4} \right\rceil \cdot 1 = 1$

$R_2^{(1)} = 1 + 1 = 2$, $I_2^{(1)} = \left\lceil \frac{2}{4} \right\rceil \cdot 1 = 1$

$R_2^{(2)} = 1 + 1 = 2 \to$ FINE $\Rightarrow R_2 = 2$

$\tau_3 \to R_3 \Rightarrow R_3^{(0)} = 2$ $I_3^{(0)} = \left\lceil \frac{2}{4} \right\rceil \cdot 1 + \left\lceil \frac{2}{5} \right\rceil \cdot 1 = 2$

$R_3^{(1)} = 4$, $I_3^{(1)} = \left\lceil \frac{4}{4} \right\rceil \cdot 1 + \left\lceil \frac{4}{5} \right\rceil \cdot 1 = 2$

$R_3^{(2)} = 4 \to$ FINE $\Rightarrow R_3 = 4$

$\tau_4 \to R_4 \Rightarrow R_4^{(0)} = 1$ $I_4^{(0)} = \left\lceil \frac{1}{4} \right\rceil \cdot 1 + \left\lceil \frac{1}{5} \right\rceil \cdot 1 + \left\lceil \frac{1}{6} \right\rceil \cdot 2 = 4$

$R_4^{(1)} = 5$ $I_4^{(1)} = \left\lceil \frac{5}{4} \right\rceil \cdot 1 + \left\lceil \frac{5}{5} \right\rceil \cdot 1 + \left\lceil \frac{5}{6} \right\rceil \cdot 2 = 5$

$R_4^{(2)} = 6$ $I_4^{(2)} = \left\lceil \frac{6}{4} \right\rceil \cdot 1 + \left\lceil \frac{6}{5} \right\rceil \cdot 1 + \left\lceil \frac{6}{6} \right\rceil \cdot 2 = 6$

9

Mo Tu We Th Fr Sa Su

No. SISTEMI INF.

Date 11 . 10 . 19

$R_4^{(3)} = 7 \quad I_4^{(3)} = 9$

$R_4^{(4)} = 10 \quad \ldots$